

IMPROVED ALGORITHM FOR DYNAMIC MEMORY LOCATION IN CLOUD COMPUTING

ABHINAV¹, NIKHIL KHANNA², NISHANT BEHAR³ & NISHI YADAV⁴

^{1,2}Department of Computer Science & Engineering, Central University, ITGGV, Bilaspur, Chhattisgarh, India

^{3,4}Assistant Professor, Department of Computer Science & Engineering,

Central University, ITGGV, Bilaspur, Chhattisgarh, India

ABSTRACT

Cloud is a new age technology used to connect to applications and data from anywhere round the globe via the internet. Anything and everything ranging from mobile phones to servers can be connected to the cloud. This has resulted into companies providing a large range of services to consumers. Upcoming businesses can now focus on its major purpose and not have to install their own servers. They can easily outsource it to these cloud providers saving them from a lot of capital investment. With the ever increasing devices on the cloud a major issue has emerged, something that needs immediate attention - Resource utilization. All the resources connected to the cloud need to function optimally for making cloud providers earn proper Return on Investment (RoI) and making the cloud users experience worthwhile. This paper proposes a new Resource Allocation Strategy for provisioning of virtual machines (VM). This work focuses on two phases, firstly the initial allocation of VMs using K-Mean Clustering technique and secondly on maximizing the optimal functioning of these VMs by migrating them from one physical machine (PM) to another. For this purpose the Load Balancing Algorithm, Skewness Algorithm and the Green Computing Algorithm are used. It also introduces a change in the Load Balancing Equation and the Skewness Equation which gives much better results.

KEYWORDS: Virtualization, Virtual Machine, K-Mean Clustering, Migration, Load Balancing Algorithm, Skewness Algorithm, Green Computing Algorithm

Original Article

Received: Apr 13 2016; **Accepted:** Apr 26 2016; **Published:** Apr 30 2016; **Paper Id.:** IJCSEITRJUN201604

I. INTRODUCTION

Cloud Computing is everywhere. It gives users access to programs and data from anywhere over the internet. The user does not need to install these applications or store the data on the local drives. Figure 1 shows some of the devices that are connected to the cloud [8].

Cloud computing provides a “computing-as-a-service” model in which compute resources are made available as a utility service—an illusion of availability of as much as demanded by the user. Moreover, users of cloud services pay only for the amount of resources (a “pay-as-use” model) used by them. This model is quite different from earlier infrastructure models, where enterprises would invest huge amounts of money in building their own computing infrastructure. Typically, traditional data centers are provisioned to meet the peak demand, which results in wastage of resources during non-peak periods. To alleviate the above problem, modern-day data centers are shifting to the cloud. [2]

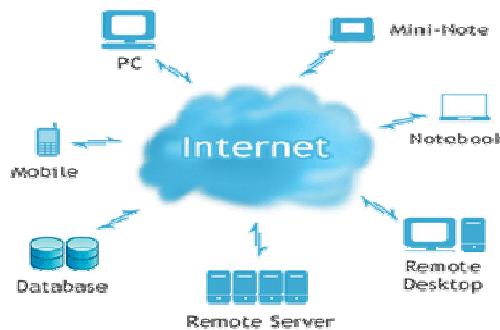


Figure 1: Everything on the Cloud [8]

Cloud requires a great deal of flexibility and agility. Dynamic scaling and shrinking is a major benefit of cloud computing for which purpose resources need to be made available at a very short notice. The cloud providers come to a Service Level Agreement (SLA) with the cloud users [5]. They need the resources to work at an optimal level to meet the requirement of users. The virtualization technology can provide these flexibilities.

After the provisioning of resources it needs to be checked whether the resources are being over utilized or underutilized. If any of these circumstances arise, the resources need to be reallocated by migrating from one physical server to another with minimal interruptions. Virtualization is one of the most commonly used solutions for resource provisioning [3]. Virtual Machine Monitor (VMM) is the technology that manages the physical allocation of virtual machines.

Changes in the resource requirements by the user can lead to the formation of cold spots and hot spots. A cold spot is a situation which arises when the resources are being underutilised. Keeping in mind the power requirements needed for cooling down the resource it is advisable to switch down such a server to help in conservation of energy. Similarly a hot spot is created when the resource is over utilised and to prevent this it is required to migrate the VM from this physical server to another server which is not itself a hotspot. Table 1 shows when to migrate from one PM to another [2].

Table 1: Dimensions of Migration

Goals	When to Migrate	Which VM to Migrate	Where to Migrate
Server Consolidation	Large number of underutilized PMs	VMs from the lightly loaded PMs	More loaded designated PMs
Load Balancing	Imbalance in the load of PMs	VMs from the overloaded PMs	Lightly loaded PMs
Hot Spot Mitigation	Resource requirements of VMs not locally fulfilled	Set of VMs from the hotspot-host	PMs which can accommodate the incoming VMs without causing new hotspots

This paper provides a Resource Allocation Strategy [1] for efficient Dynamic Memory Allocation which has been explained in detail in Section III of this paper. Section II provides details of the related works which we have studied to propose our algorithm.

II. RELATED WORKS

Dynamic VM Allocation using Clustering Algorithm

Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. In [4], it is proposed to implement K-Means clustering algorithm in cloud computing environment. It is obtained that both Data Mining techniques and Cloud Computing helps the business organizations to achieve maximized profit and cut costs

in different possible ways. Thus K-Means clustering algorithm, which is one of the very popular and high performance clustering algorithms, is used in cloud.

Fast and accurate k-means clustering is done for large datasets. As cloud computing deals with large data centres, so huge amount of data need to be accessed simultaneously. Thus it impacts on the performance. In this way, fast and accurate clustering algorithm seems to be a better option to adopt in clouds. It has been seen that K-Means clustering is much faster than other Divide and Conquer Algorithms.

K-Means follows the partitioned or non-hierarchical clustering approach. It involves partitioning the given data set into specific number groups called Clusters. Each cluster is associated with a centre point called centroid. Each point is assigned to a cluster with the closest centroid.

Algorithm

Input: List V of Virtual Machine`s with their location around the globe List D of datacenters

- Select K points according to the number of datacenters in D
- Choose datacenter from D
- Form K clusters of VM`s from V by assigning closest centroid
- Recomputed the centroid of each cluster
- Arrange all the requested VM`s in cluster form
- Allocate the VM`s to the available Host
- If all the VM`s are allocated
- Assign the VM`s cluster to the selected datacenter
- End if
- Repeat [2] until D is empty
- If all the VM`s are created in the datacenters
- Send the cloudlets to the created VM`s
- End if
- Compute the results

The initial centroid will be chosen randomly. The centroid is nothing but the mean of the points in the cluster. Euclidean distance is used to measure the closeness. K-Means generates different clusters in different runs.

Load Prediction Equation

In [6], the future memory demand is predicted using exponentially weighted moving average (EWMA) using a TCP-like scheme:

$$E(t) = \alpha * E(t - 1) + (1 - \alpha) * O(t - 1)$$

E (t): Estimated memory usage at time t

E (t-1): Estimated memory usage of past time

O (t-1): Observed memory usage of past time

$$\alpha = \frac{O(t-1)}{E(t-1)}$$

α - Denote the trade-off between the observed and estimated value

Future demands are forecasted to ensure the memory is properly managed and only unwanted memory area is released.

Skewness Equation

In [6], the concept of “skewness” is used to measure the unevenness in the multi-dimensional resource utilization of a server. By minimizing skewness, one can combine different types of workloads nicely and improve the overall utilization of server resources.

One can use skewness algorithm to ensure there is uniform utilization of memory across VMs.

$$P = r/R - 1$$

Where, p - skewness to determine the usage level of memory by a VM

r - Memory usage of the current VM under evaluation

R - Average of memory usage by all VMs

If $P = 0$, the Memory utilization is normal

$P > 0$, then the memory usage is more compared to others VMs

$P < 0$, then the memory usage is less compared to other VMs

Green Computing Algorithm

When the resource utilization of active servers is too low, some of them can be turned off to save energy. This is handled by the green computing algorithm. [6]

Threshold limits are set for hot, warm and cold spot values as 85, 60 and 25.

- Calculate the % of memory usage based on the predicted value
- Check against the threshold limit to check any VM falling under hot spot
- If the VM come under hot spot, call page replacement algorithm
- If the VM fall under cold spot, add it to balloon list with % free details
- If the VM fall under warm temperature – No action
- When the VM is in cold spot we make entry in balloon list.

III. PROPOSED ALGORITHM

The proposed algorithm focuses on a strategy to utilize a set of techniques together to achieve higher levels of functionality for the VMs. We have also introduced certain changes to a couple of these techniques which has led to better results.

In the proposed strategy we have firstly allocated memory to the VMs using the K-Mean Clustering algorithm. In the later stage we focus on reaching optimal functioning of these VMs by applying on them other algorithms one after another. Various parameters used in simulating the proposed algorithm is shown in Table 2.

Table 2: Simulation Parameters used in the Algorithm

Notations	Description
$E(t)$	Estimated memory usage at time t
$E(t-1)$	Estimated memory usage of past time
$O(t-1)$	Observed memory usage of past time
α	Denote the trade-off between the observed and estimated value
r_i	Utilization of the i -th resource
R	Average utilization of all resource
p	Skewness

We begin by discussing the various changes incurred by us to these techniques.

Load Prediction Equation

$$E(t) = \alpha * E(t-1) + |(1 - \alpha)| * O(t-1)$$

$E(t)$ – Estimated memory usage at time t

$E(t-1)$ – Estimated memory usage of past time

$O(t-1)$ – Observed memory usage of past time

$$\alpha = \frac{O(t-1)}{E(t-1)}$$

α - Denote the trade-off between the observed and estimated value

The proposed change is the addition of the modulus operator to the term $|(1 - \alpha)|$ in the equation. This prevents the estimated value of memory from going to the negative region as there is no such thing as a negative memory.

Skewness Equation

Let n be the number of resources we consider and r_i be the utilization of the i -th resource. We define the resource skewness of a server p as

$$\text{skewness}(p) = \sqrt{\sum_{i=1}^n \left(\frac{r_i}{R} - 1 \right)^2}$$

Where R is the average utilization of all resources for server p [7].

The proposed change in the equation for skewness involves the squaring and square rooting of the previous equation. This change helps improve the net value of skewness as negative values tend to cancel out the positive values resulting in wrong data.

By minimizing the skewness, we can combine different types of workloads nicely and improve the overall utilization of server resources.

After bringing on the changes proposed to these equations, we propose our algorithm.

Proposed Algorithm

- Perform K-Mean Clustering on VMs
- Perform Load Balancing Algorithm for each VM
- Perform Skewness Algorithm for each VM
- If VM is over utilized or under-utilized Then
- Perform Green Computing Algorithm for each VM
- End If
- If VM requirement changes
- Repeat [2]
- End if
- Repeat [1] when new VMs need to be allocated

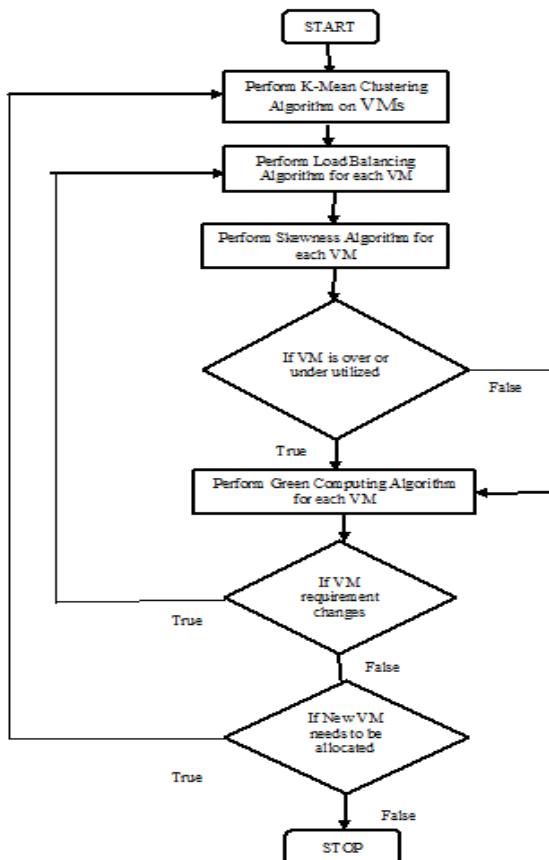


Figure 2: Flowchart for Proposed Algorithm

IV. SIMULATION RESULTS

Simulation results are obtained using C++ programming. And the results are plotted on graphs so as to see the drastic changes in results obtained. Both in Figure 3 as well as in Figure 4 the red line denotes the results obtained by the implementation of previous equations and in the blue line denotes the changes observed by the proposed equations.

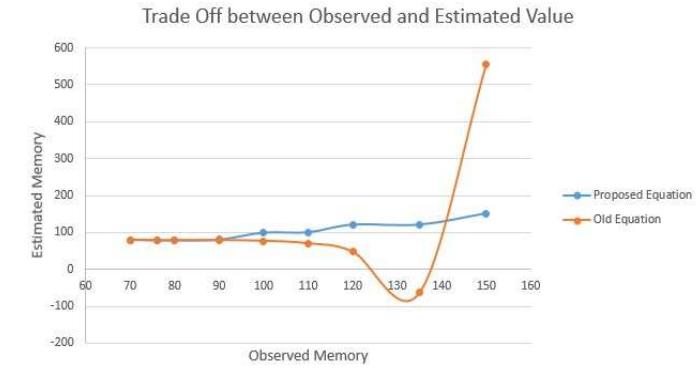


Figure 3: Change in the Graph for EWMA

From Figure 3 we see how the earlier equation used to guess the correct estimated memory usage for small changes but used to take three to four attempts before correct guessing the value for larger changes. We also notice the drastic improvement in the results with the inclusion of the modulus operator. The false estimations have reduced to almost none.

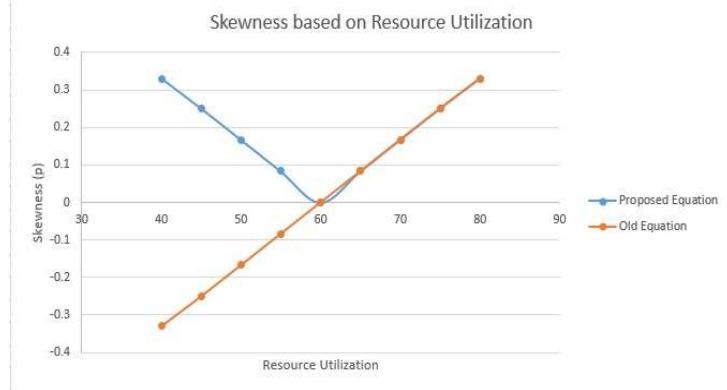


Figure 4: Change in the Graph for Skewness

From Figure 4 it can be seen that the old equation used to give negative value of skewness which is not false for individual resource but it did not take into consideration the changes due to other resources. With the implementation of the proposed equation the overall deviations in the skewness of each VM is taken into consideration.

V. CONCLUSIONS AND FUTURE WORK

From the simulation results, it can be concluded that our proposed changes in the Load Balancing Equation and the Skewness Equation results in better judgement of the functioning of the resource namely the memory. This results in the reallocation of memory pages of a VM from one PM to another without hindering the performance of the cloud services. The proposed work will help reduce the cost of providing services to the cloud clients and improve access time for the resources, which will benefit both the service provider and the customers.

REFERENCES

1. V. Vinothina, Dr. R. Sridaran, Dr. PadmavathiGanpathi, "A Survey on Resource Allocation Strategies in Cloud Computing", *IJACSA* Vol 3, No. 6 2012.
2. Mayank Mishra, Anvesha Das, Purushottam Kulkarni, AnirudhaSahoo, "Dynamic Resource Management Using Virtual Machine Migration", *IEEE Communication Magazine*, September 2012
3. N. Krishnaveni, G. Sivakumar, "Survey on Dynamic Resource Allocation in Cloud Computing Environment", *IJCATR*, Vol 2 Issue 6 pp 731-737,2013
4. Bhupendra Panchal, R. K. Kapoor, "Dynamic VM Allocation Algorithm using Clustering in Cloud Computing", *IJARCSSE* Vol 3 Issue 9 September 2013
5. S. R. Suraj, R. Natchadalingam, "Adaptive Genetic Algorithm for Efficient Resource Management in Cloud Computing", *IJETAE* Vol 4 Issue 2 February 2014
6. V.Holy Angel Jenitha, R. Veeramani, "Dynamic Memory Allocation using Ballooning and Virtualization in Cloud Computing", *IOSR-JCE* Vol 16 Issue 2 Version 2 (March April 2014) pp 19-23.
7. Seematai S. Patil, KogantiBhavani, "Dynamic Resource Management Using Virtual Machine for Cloud Computing Algorithm", *IJEAT*, Vol 3 Issue 6, August 2014
8. Retrieved from http://warrenmars.com/soap_box/computers/beware_google.htm

AUTHOR DETAILS

- ABHINAV is a Student of B.Tech, Computer Science & Engineering, Central University, ITGGV, Bilaspur (Chhattisgarh), INDIA
- NIKHIL KHANNA is a Student of B.Tech, Computer Science & Engineering, Central University, ITGGV, Bilaspur (Chhattisgarh), INDIA
- NISHANT BEHAR is Assistant Professor at Department of Computer Science & Engineering, Central University, ITGGV, Bilaspur (Chhattisgarh), INDIA.
- NISHI YADAV is Assistant Professor at Department of Computer Science & Engineering, Central University, ITGGV, Bilaspur (Chhattisgarh), INDIA.